Joonas Forsberg

Implementation of Centralized Log Management Solution for Ensuring Privacy of Individuals as Required by EU Regulation

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

5 March 2018

Metropolia

Author	Joonas Forsberg
Title	Implementation of Centralized Log Management Solution for Ensuring Privacy of Individuals as Required by EU Regulation
Number of Pages Date	49 pages + 2 Appendices 2 March 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	
Instructors	Ville Laitinen, Senior Consultant Henrik Rak, Legal Counsel Kimmo Saurén, Senior Lecturer

The primary purpose of this thesis was to explore the new European Union (EU) data privacy regulation, General Data Protection Regulation (GDPR), to ensure the case company is compliant with the regulation when it comes into effect on 25th of May 2018. The regulation requires organizations to implement necessary technical solutions and organizational processes to ensure the privacy of individuals is not violated.

The secondary purpose of the thesis is to research System Information and Event Management (SIEM) solutions, which can be used to cover the technical requirements set by the regulation. Commercial SIEM solutions are often designed to meet the necessary auditing requirements set by the regulation, but are also rather heavyweight to be implemented for smaller organizations and therefore a log management solution, Elastic Stack, was looked at in detail.

The outcome of the thesis was a functional Elastic Stack deployment, which meets the basic functionality of a SIEM solution. With the system, it is possible to demonstrate that the client company processes personal information in compliance with the regulation and is capable of noticing security threats as required by the regulation.

Because of the thesis, the personal data protection and the level of security of the IT environment has been increased to match the challenges set by modern security threats in regards to log management and ability to define the scale of security breach. The thesis was carried out from the case company's point of view, but can be utilized by small and medium-sized businesses that are not focused on dealing with personal information.

Keywords	2016/679 (GDPR), SIEM, Elastic Stack, Information Security, Log Management

Tekijä	Joonas Forsberg
Otsikko	Keskitetyn lokienhallintajärjestelmäratkaisun toteuttaminen henkilöiden yksityisyydensuojan turvaamiseksi EU-asetuksen mukaisesti
Sivumäärä Päivämäärä	49 sivua + 2 liitettä 2.3.2018
Tutkinto	Bachelor of Engineering
Koulutusohjelma	Information Technology
Suuntautumisvaihtoehto	
Ohjaajat	Senior Consultant Ville Laitinen Legal Counsel Henrik Rak Lehtori Kimmo Saurén

Insinöörityön pääasiallinen tarkoitus oli tutkia uutta Euroopan unionin tietosuoja-asetusta (General Data Protection Regulation, GDPR) ja varmistaa, että työn tilaajayritys täyttää asetuksen vaatimukset asetuksen tullessa voimaan 25.5.2018. Asetus velvoittaa organisaatioita luomaan tarvittavat tekniset ratkaisut ja organisaatioprosessit yksilöiden tietoturvan turvaamiseksi.

Insinöörityön toisena tarkoituksena oli tutkia Security Information and Event Management (SIEM) -ratkaisuja, joita on mahdollista käyttää hyväksi asetuksen teknisten vaatimusten täyttämiseksi. Kaupalliset SIEM-ratkaisut on lähtökohtaisesti suunniteltu siten, että niiden auditointiominaisuudet täyttävät asetuksen asettamat vaatimukset. Nämä järjestelmät ovat kuitenkin useimmiten liian raskaita pienille organisaatioille, joten Elastic Stack -nimistä lokienhallintajärjestelmäratkaisua tutkittiin työssä tarkemmin.

Insinöörityössä toteutettiin Elastic Stack -lokienhallintajärjestelmäratkaisu, joka täyttää SIEM-järjestelmän perustoiminnallisuudet. Järjestelmän avulla on mahdollista näyttää toteen, että työn tilaajayritys käsittelee henkilötietoja GDPR:n asettamien vaatimusten mukaisesti ja kykenee havaitsemaan tietoturvauhkia asetuksen vaatimalla tavalla.

Insinöörityön ansiosta tilaajayrityksen henkilötietosuoja sekä palvelin- ja laiteympäristön tietoturvan taso on saatu vastaamaan nykyaikaisten tietoturvauhkien asettamia haasteita lokitietojen keräämisen ja tieturvapoikkeuksien laajuuden selvittämisen osalta. Insinöörityö tehtiin tilaajayrityksen näkökulmasta, mutta se on lisäksi sovellettavissa pienille- ja keskisuurille yrityksille, joiden ydinliiketoimintaan ei kuulu henkilötietojen käsittely.

Avainsanat	2016/679 (GDPR), SIEM, Elastic Stack, Tietoturva, Lokienhallinta

Metropolia

Contents

1 Introduction 1

2 The General Data Protection Regulation 3

2.1 Background 3
2.2 Basic Principles of GDPR 4
2.2.1 The Definition of Personal Data 6
2.3 Rights of Data Subjects 7
2.4 Accountability and Obligations 8
2.5 Regulation Compliance 10

3 GDPR in Practice at NAPA 12

3.1 Data Processing 12
3.2 Data Classification and Policies 13
3.3 Legacy Systems, File Shares and Data Archives 14
3.4 Preparations for GDPR Compliance at NAPA 15

4 Security Information and Event Management 17

4.1 Overview of SIEM 17
4.2 Defining Features of SIEM 19
4.3 SIEM Solutions 21
4.4 SIEM in Relation to GDPR 23

5 Elastic Stack 25

5.1 Core Components 25
5.1.1 Elasticsearch 25
5.1.2 Logstash 29
5.1.3 Kibana 30
5.1.4 Beats Platform 32
5.1.5 X-Pack 33
5.2 Availability and Performance 33
5.3 Elastic Stack as a Full-scale SIEM? 35
5.4 Considerations for the Implementation 36

6 Elastic Stack Implementation 39

6.1 Elastic Stack Preparations 39
6.2 Preparing the IT Environment at NAPA for the Elastic Stack 40
6.2.1 Configuration Management 42

6.3 Elastic Stack Configuration and Security 43
6.4 M-Files Event Log Processing 45

7 Conclusions

References

Appendices

Appendix 1: Source code (C#) of the M-Files log extractor prototype

Appendix 2: Logstash pipeline configuration file for M-Files event logs

Abbreviations

AD	Active Directory
AIS	Automatic Identification System
API	Application Programming Interface
CA	Certificate Authority
CPU	Central Processing Unit
CRM	Customer Relationship Management
DHCP	Dynamic Host Configuration Protocol
DPA	Data Protection Authority
DPO	Data Protection Officer
ECM	Enterprise Content Management
ERP	Enterprise Resource Planning
EU	European Union
GDPR	General Data Protection Regulation
GPS	Global Positioning System
HIDS	Host-based Intrusion Detection system
HR	Human Resources
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IT	Information Technology
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
NRT	Near Real Time
OECD	Organisation for Economic Co-operation and Development
OOP	Ordinary Object Pointers
PCI-DSS	Payment Card Industry Data Security Standard
REST	Representational State Transfer
RFC	Request for Comments
SAAS	Software as a Service
SIEM	System Information and Event Monitoring
SQL	Structured Query Language
TLS	Transport Layer Security
URL	Uniform Resource Locator
WMI	Windows Management Instrumentation
VPN	Virtual Private Network
XML	Extensible Markup Language

1 Introduction

This study explores the *General Data Protection Regulation* (*GDPR*), which is an EU regulation (2016/679) with a purpose to strengthen and unify the data protection of individuals residing within the EU. The regulation was launched on 27th of April 2016 and it becomes effective after a 2-year transition period, on 25 of May 2018.

Due to the regulation, companies and organizations that are active within the EU, must handle all personal data of all natural persons residing in any EU member state according to the regulation. This thesis focuses on the implementation of a SIEM solution for ensuring the privacy of individuals as required by EU regulation.

This thesis is carried out for Napa Oy, later referred to as NAPA, with the primary purpose of helping the company to become compliant with GDPR by preparing technical solutions to monitor the usage of information systems containing personal data. The secondary purpose of this thesis is to strengthen the overall security of the information systems. The implementation of a SIEM system plays a key role in both objectives as it provides a system, which together with improved internal policies, can be used to ensure and prove compliance with GDPR.

NAPA is a global market-leading software company, providing design and operation solutions for the global maritime industry. NAPA has employees in 10 different countries and more than 700 user organizations globally. The personal data processed at NAPA are mostly related to their own employees and consists of information linked to employment, such as payroll records and other private personal information.

Some of the personal data related to an employee could be used to conduct an identity theft or otherwise cause a serious harm to the employee, which makes it important to implement necessary data protection practices to safeguard the data, which is directly linked to an individual. The regulation also enforces organizations to implement necessary technical solutions to monitor and detect data breaches, for which the SIEM solution is used for.

This thesis is structured in seven sections and the first one contains a brief introduction to the subject of data protection of natural persons residing in any EU member state, and to the case company NAPA. Section two focuses on the GDPR, explaining what it is in

detail and contains information about the obligations the GDPR sets for organizations when handling personal data that is governed by the GDPR. The third section approaches the GDPR from the point of view of the case company, essentially providing an answer to a fundamental question: what does NAPA need to do before the regulation comes into effect?

The fourth section provides a basis for the technical platform to be implemented in later sections by focusing on the concept of SIEM and how it relates to GDPR. The fifth section contains theoretical background of the Elastic Stack, which is the solution that is implemented in section six. Section seven contains conclusions of the implementation process and recommendations for further actions that should be taken care of before the regulation comes into effect.

2 The General Data Protection Regulation

GDPR will supersede the data protection directive (95/46/EC), which has been in use since the year 1995. The difference between a regulation and directive is that regulations are enforced without the need of the governments to implement legislations to enable them, unlike directives, which require a local legislation in order to be effective. (Council Regulation (EU) 2016/679 2016: 1.)

GDPR also applies to companies, which have no legal establishments within the EU, if they are processing any data related to any natural person residing in a EU member state. This is a major difference between the former directive and the new regulation, enforcing companies globally to review their business processes, practices and contracts. (Council Regulation (EU) 2016/679 2016: 32-33.)

2.1 Background

Both GDPR and the data protection directive are based on the guidelines and instructions published by the Organisation for Economic Co-operation and Development (OECD) in the year 1980. While the guidelines were published in an era in which most of the data was stored on a physical medium, most of the guidelines are still true today. However, with the adaptation of social media and various cloud-based services, the guidelines needed to be updated. (eugdpr.org 2016.)

The OECD guidelines were a set of recommendations endorsed by both the US and EU to protect personal data and create a fundamental human right of privacy. The guidelines included the same themes as the content of GDPR today, such as the limitations to only use the data for a specific purpose and safeguarding the data against unauthorized access or modification. (eugdpr.org 2016.)

The data protection directive harmonized data protection laws across the EU and implemented rules of data transfer to countries outside of the EU. One of the major accomplishments of the directive was the formation of the Data Protection Authorities (DPAs) in all EU member states to oversee the implementation of the directive and provide a physical manifestation to interact between businesses and residents of member states. (eugdpr.org 2016.)

However, as only 1% of the total population in EU was using the internet at the time, the directive does not meet the requirements of the modern world where businesses are handling more personal data than ever before (eugdpr.org 2016). Furthermore, the amount of data is expected to keep rising, as more and more services will be available on the internet (Georges 2017).

2.2 Basic Principles of GDPR

The regulation categorizes the persons to whom the regulation applies to as data controllers and data processors. By definition, data controllers define how and why the data is processed and the data processor acts on behalf of the controller. The data processor is obligated to store and maintain proper records of any personal data and activities performed on the data. Controllers, on the other hand, are responsible for ensuring that the contracts with the data processors are in compliance with the regulation. (Information Commissioner's Office (UK) 2017.)

The regulation is not applied if the data processing is covered by the Law Enforcement Directive (EU) 2016/680, is done for the purposes of national security or by individuals for personal activities. (Information Commissioner's Office (UK) 2017.)

The regulation applies to any data, which can either be classified as personal data or sensitive personal data. Personal data could be any data, which can be linked to any identified or identifiable natural person (“data subject”). (Gabel & Hickman 2016.) Examples of such data are contact details, identification numbers, location data, personal information, genetic, cultural or any online identifiers, such as screen-names.

Sensitive personal data is defined in the regulation as:

> data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation (Council Regulation (EU) 2016/679: 38).

One of the major themes in the regulation is the concept of lawful processing, which means data controllers and processors must be able to identify a lawful basis before they

can start to collect and process any personal data. The lawful bases for processing of personal data must be defined and documented in a proper manner. The regulation specifies the lawful bases which can be used for processing personal data and special categories of data, which are also defined in the regulation. (Information Commissioner's Office (UK) 2017.)

Consent is one of the key lawful bases, which are providing the possibility to collect and process the data of individuals. Under the regulation, consent should be given freely, it should be specific and provide explicit indication of individuals' wishes. Individuals must opt-in and the consent cannot be acquired by implementing methods, which opt-in the individuals without their knowledge or acknowledgement. Consent must be separated from the terms of usage and when asking for a consent, the statement of consent must be written clearly and as accurately as possible, meaning organisations cannot use a blanket-statement to enable data collection. (Information Commissioner's Office (UK) 2017.)

Data processors must be able to verify the consent given by individuals and there should be an effortless way to withdraw consent if so desired. Data processors are not allowed to deny the withdrawal of consent if the data subject requests it. (Information Commissioner's Office (UK) 2017.)

There are also cases that are explicitly defined in the regulation, in which data can be collected without receiving a consent from the data subject. For example, the regulation allows organization to process personal data without receiving consent from the data subject, if the processing is necessary due to legal obligations or is done to protect the vital interests of any natural person. (Council Regulation (EU) 2016/679 2016: 36, 38.)

To ensure local laws and regulations are considered when implementing GDPR on a national level, a supervisory authority will be established in each EU member state. In Finland, the supervisory authority is Data Protection Authority ("tietosuojavirasto") and they will take over the current activities of the Data Protection Ombudsman ("tietosuoja-valtuutetun toimisto"), in addition to official duties required by the regulation. Local legislation in Finland is still work in progress and once completed, it will supplement the regulation on topics which the regulation does not account for. (Lång et al., 2017.)

Sanctions for not complying with the regulation have been the most debated topic of the regulation, as the supervisory authorities are given the power to issue administrative fines up to a maximum of 20 million euros or 4% of annual turnover of the previous fiscal year, whichever is greater. The authority can also issue warnings, orders and reprimands, which should be used as a primary tool when dealing with GDPR infringements. (Loyens & Loeff 2017.)

When issuing fines, the supervisory authority must consider the severity, nature, type of negligence and previous sanctions, when deciding about issuing the administrative fines to a company. The sanction must be in line with the severity of the infringement (Loyens & Loeff 2017), which means in practice only serious and continuous violations will result in an administrative fine.

2.2.1 The Definition of Personal Data

In the article 4(1) of the regulation (Council Regulation (EU) 2016/679 2016: 33), personal data is defined as:

> any information relating to an identified or identifiable natural person (“data subject”).

For a dataset to be considered personal data, it must contain an identifier, which is a piece of any information that enables identification of an individual (Council Regulation (EU) 2016/679 2016: 33). What an identifier is, has not been formally defined in the regulation, but the regulation sums up common identifiers and broad categories of different identifiers which can help with the definition.

The article 4(1) provides a few examples of identifiers as:

> name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity (Council Regulation (EU) 2016/679 2016: 36).

The recital (30) of the regulation further clarifies the definition of identifier by stating that the following information can be associated with an individual to create an identifier:

> devices, applications, tools and protocols, such as internet protocol addresses, cookie identifiers or other identifiers such as radio frequency identification tags (Council Regulation (EU) 2016/679 2016: 6).

The definition of an identifier is not formally defined in the regulation, which means the list of online identifiers can be continued with a vast amount of data elements, which could be used to enable the identification of an individual. Due to the definition of personal data in the regulation, it is necessary to take the regulation literally when dealing with information that might be considered to be personal data. (i-SCOOP 2017.)

2.3 Rights of Data Subjects

The regulation provides the data subjects eight basic rights which provide them better control over their personal data.

The right to be informed obligates data holder to provide information about how and which data is processed. This is usually done by privacy notice in which the content should be transparent and easy to understand. (Information Commissioner's Office (UK) 2017.)

The right of access provides data subjects the right to obtain confirmation about the processing of the data gathered, access to their personal data and supplementary information free of charge (Council Regulation (EU) 2016/679 2016: 43).

The right to rectification allows data subjects to demand rectification of data, in case the data is incomplete or otherwise inaccurate. If the data holder has disclosed the information with a third party, the data holder is obligated to inform any third parties about the rectification. (Information Commissioner's Office (UK) 2017.)

The right to erasure enables individuals to request the removal of personal data from the information systems. The right does not provide an absolute right for data removal and it applies only in specific circumstances or when the organization is unable to provide a proper reason to continue the processing activities. (Information Commissioner's Office (UK) 2017.)

The right to restrict processing allows the data subjects to prevent data holders from further processing the collected data. The personal data is allowed to be stored and enough data can be retained to ensure no further processing is required. (Information Commissioner's Office (UK) 2017.)

The right to data portability provides the individuals the right to obtain data related to them from the data holder and reuse it any purpose of their own. The data should be possible to transfer, move or copy from one system to another without compromising the security or affecting the usability of the data. (Information Commissioner's Office (UK) 2017.)

The right to object is giving the data subjects a right to object to processing the data for scientific/historical research purposes, direct marketing or data processing in general. The data holders must stop the data processing unless they can show a legitimate reason for data processing or the data is processed in defence of any legal claims. (Information Commissioner's Office (UK) 2017.)

Rights related to automated decision making and profiling provides data subject with protection for damages or legal effects caused by automated decision making, in which a human interaction has not been made. (Information Commissioner's Office (UK) 2017.)

For the purposes of this thesis, the most relevant rights are the right of access, the right of erasure and the right to data portability, as the other rights have no significant impact in the current way of data processing at NAPA.

2.4 Accountability and Obligations

Article 5(2) of the regulation introduces an accountability principle, which defines the data controller as the responsible party who has to be able to prove compliance with the article 5(1), which defines the principals related to the processing of personal data (Council Regulation (EU) 2016/679 2016: 35-36).

Article 24 defines the responsibilities of the controller in detail, which sets the basis on how to approach the regulation. The controller must have proper technical and organi-

zational measures in place, which ensure the processing follows the regulation. The article also obligates the data controller to implement appropriate data protection policies, which could include concepts such as data retention, usage and encryption policies. (Council Regulation (EU) 2016/679 2016: 47.)

Data controllers can demonstrate the compliance by implementing technical solutions as stated in article 5(2) and maintaining appropriate documentation of processing activities, including details about the scope of data, data source, reasons for data processing and list of persons who are responsible for the data processing. The regulation implements an obligation to implement privacy by design and by default. (Information Commissioner's Office (UK) 2017.)

This means that the regulation in practice enforces companies to implement technical solutions and organisation processes, which are demonstrating the data protection is integrated into the data processing activities. In companies and organizations, data protection should be considered in the design phase of a data management system, to ensure data protection is enabled by default and privacy concerns are accounted for.

Data gathered from the data subjects should be kept to a bare minimum needed to complete any given objective and when possible, the data should be pseudonymized to avoid unnecessary relation to the data subjects (Information Commissioner's Office (UK) 2017). For example, the processing of information about individual salaries can be conducted without lawful reason, if the information processed cannot be traced back to an individual person.

Transparency is a crucial factor of compliance, as the data subjects have a right to access the data collected from them and be informed on how the data is processed. Individuals could be allowed to monitor the processing activities or automatically download all the data which have been collected and is being used for the data processing. (Information Commissioner's Office (UK) 2017.)

Companies that have more than 250 employees are enforced to maintain additional records of the processing activities. Companies with less than 250 employees are only required to maintain records that are associated with a higher risk, such as the processing of personal data, which might lead into to risk of data subject’s rights or freedom; or

processing of data which have been classified as a sensitive personal data. (Information Commissioner's Office (UK) 2017.)
The records maintained must store information about the organization processing the data, name of the data protection officer (DPO), purpose of processing, descriptions of the categorisation of personal data, details of transfers to third countries or parties including necessary information about the mechanisms of transfer and safeguards in place during the transfer. The records must also have information about retention policies and description of security measures related to the data processing. (Information Commissioner's Office (UK) 2017.)

While the records are considered internal and are not to be shared with third parties, they must be presented to the supervisory authority in case of an ongoing investigation in which the records are needed (Information Commissioner's Office (UK) 2017).

2.5 Regulation Compliance

The regulation does not contain a distinct list of the security requirements, which are needed to be implemented to ensure compliance with the regulation. Instead, they are spread throughout the regulation in various chapters and articles. (Gemalto 2017.)

Data encryption and the limitation of access are one of the basic principles of securing the data mentioned in the regulation, as the data should be encrypted and therefore unreadable by anyone who is not authorized to view the data. (Gemalto 2017.)

As the data should only be processed by authorized personnel, some sort of a monitoring system should be implemented to keep track of which information is being accessed by whom. The system also helps in providing a means of identifying any unauthorized data modification by authorized or unauthorized personnel, which in turn helps to ensure the integrity and accuracy of the data.

If an individual decides to revoke their consent based on the right to erasure, an organization must completely erase any data, which are related to the said data subject. From a technical point of view, this is achieved by encrypting the data and disposing of the encryption key. (Gemalto 2017.) However, to begin the erasure process, the organization must be aware of all the locations in which the data is located. If any information is moved

or replicated to a location different from the original location, an event should be generated. Events are single occurrences in systems, such as a single line in a log file, or a single Hypertext Transfer Protocol (HTTP) request.
Organizations also need to mitigate the risks of data exposure and perform due diligence, by implementing measures to ensure and demonstrate compliance, conduct risk assessment and demonstrate full control of the data (Gemalto 2017).

Organizations are also responsible for notifying customers and the supervisory authority whenever a breach, which threatens the privacy of data subjects, has occurred. Organizations must notify the supervisory authority within 72 hours of noticing the breach and describe the consequences of the data breach. Organizations can avoid the notification obligations if the data affected is encrypted and proper key-management practices have been followed or if the rights of the data subject are not at risk. (Gemalto 2017.)

3 GDPR in Practice at NAPA

Regulatory compliance is important for NAPA but due to the small size of the company and the field NAPA operates at, the regulation does not pose a significant threat to the company. Accordingly, GDPR compliance can be mostly achieved through policies, documentation and practices of handling any data that could be considered personal data, such as employee records or contact information of the customers.

Internal policies, documentation and practices are excluded of the scope of this thesis and instead the thesis focuses on the technical aspect of the regulation. While the processes related to the management of any personal data is an important aspect of GDPR compliance, the processes are not closely related to the technical solution used to monitor the usage, and therefore it was decided to exclude them from the thesis.

The author of this thesis is at the time of the writing employed by NAPA. The content of this chapter, unless otherwise stated, is based on observations made by the author, internal research and discussions with other employees at NAPA.

3.1 Data Processing

The personal data that is processed at NAPA is mostly limited to information about employees and subcontractors, such as payroll information and work agreements. Customer related information, such as names and e-mail addresses, are collected in small quantities as well. Additional customer related information, such as login addresses and usernames, are gathered from the NAPA Software as a Service (SaaS) solution.

During the last couple of years, NAPA has taken into use several tools, such as Enterprise Content Management (ECM) and an Enterprise Resource Planning (ERP) system, both of which have a crucial part when ensuring compliance with GDPR.

The ERP system has replaced most of the legacy systems, which had been previously used for various tasks, such as financial reporting, Human Resource (HR) management and customer relationship management (CRM), and most of the processing of personal data related to customers and employees is done inside of the system. Some information, such as agreements and other documents, are stored and managed in M-Files,

which is the ECM system in question. Together these two systems form a coherent platform for managing personal data at NAPA.

The software and SaaS solutions developed by NAPA for the maritime industry are not collecting or using data which could be connected to a single user of the software. Most of the data that is collected and could be considered personal, is a result of identity management, logging functionality or usage pattern analysis. Some solutions are using tracking information such as Global Positioning System (GPS) coordinates or data from Automatic Identification System (AIS), as a part of the analysis of the ship performance, but for as long as the data gathered cannot be connected to any particular person onboard it is not considered to be personal data.

3.2 Data Classification and Policies

The types of personal data can be categorized into three distinct categories: HR, Customer and Information Technology (IT) data. Each category has a distinct level of sensitivity and therefore, the access to the data is limited based on the employee roles. Most of the customer related data are accessible by all employees of NAPA, but HR and IT data are restricted to the eyes of authorized personnel only.

HR data consists of employment, payroll and other information, which are directly linked to an employment of an individual and most often include personal information, such as personal identity numbers or taxation information. HR related data should be only viewable by HR department and any unauthorized processing of employment related information should generate an alert to notify the HR personnel about the file access.

IT data consists mostly of information related to Active Directory (AD) user accounts, which are used to manage employee identities across the various systems and services. Since the AD user accounts are always personal and shared accounts are not used, any log file that contains a username can be directly linked to a specific employee. Other information, which is not tied to a specific user account, is mostly information that can be indirectly linked to an employee, such as IP addresses, which are saved by the firewall when accessing the internal systems from a remote address.

Most of the IT data is gathered from log files from various systems and the log files should only be used to detect or investigate breaches or debug problems related to these systems. The log files should only be viewable by authorized personnel and any changes to groups providing access to the data, or the user accounts having access to the data, should generate an alert.

Customer related data that are accessible by most employees do not need a tight access control policy, as the information typically is not sensitive or could be used to cause a security incident or violate the privacy of a customer in a meaningful way.

Despite of distinctive differences, all the categories have a few things in common. For example, to prove compliance with GDPR, an event should be generated whenever data from any of the categories is accessed containing information about the data accessed, showing who, when and from where it was accessed.

If the data falls to any of the three categories or is in some other way considered personal data, it should be stored either in the ERP system or the ECM, and classified accordingly to enable further processing if needed.

3.3 Legacy Systems, File Shares and Data Archives

One of the major challenges of ensuring the compliance with the regulation is the number of legacy systems still in use and data archives, which consist of archived file shares and physical document archives, which are a result of the company's 30-year history (NAPA 2017). While most of the data can be pseudonymized or permanently deleted, there are some information which must be stored for a set amount of time due to legislation, such as payroll information and other records related to employees.

The data archives contain documents, which have not been used for quite some time and are unlikely to be actively used anytime in the future, unless an unexpected event, such as a legal action taken against the company or the ownership of an intellectual property has to be proven. The data archives are a combination of logically structured data and randomly placed unstructured data, which in most cases is not classified in any way making it nearly impossible to determine whether the data contains personal data or not.

The number of legacy systems which are still partially in use, are also a source of problems not only for GDPR compliance, but also from the information security point of view. The legacy systems are either end-of-life or are otherwise in a state in which they can no longer be patched regularly. For example, a legacy ERP system is still being used to retrieve information about old customer support cases containing contact details of the customers, which are considered to be personal information from the regulation's point of view and therefore, are subject to the requirements of the regulation.

Network shares have been traditionally used to store several types of data in a comparable way as a data archive would be used. In practice this means some of the data have been stored purely for archiving purposes and therefore, it is difficult to define whether a file should be archived or kept on the network share. Some of the data is structured, but most of the data is something, which is not structured in any way, making it hard to properly identify what sort of data is stored on the network shares and who should have access to the data.

3.4 Preparations for GDPR Compliance at NAPA

A starting point to prepare NAPA for the regulation is an introduction and training session about the fundamental idea and requirements of the regulation. All employees should be aware of the basic principles of the regulation in order for them to properly categorize any information they enter to any information system. Policies on how to manage data should be established and the existing data in the ERP system and in M-files should be categorized accordingly.

From a technical point of view, one of the first things that should be done, is to prepare the IT environment for the regulation, which in practice means migrating away from the legacy systems and shutting them down. Most of them are not only costly to maintain, but cause a security risk and are something, which would cause a lot of wasted work if they would be made compliant with the regulation.

The current file shares must also be reorganized in a way all stored data is structured and has a reason to be stored on the file server. Documents, which fall under the regulation and can be put to M-Files, should be placed there and removed from the file server. During the process of reorganization, all data should be classified either by file or folder.

In an optimal case, the file shares should not contain any information, which would fall under the regulation.

Lastly a SIEM system should be implemented to monitor any access to the files falling under the regulation and other events which might potentially expose the files to unauthorized personnel, such as modification of folder-level permissions or changes to user groups granting access to the data protected by the regulation. The implementation of the system, which provides the necessary technical solutions to ensure the compliance with the regulation is covered in the next sections.

4 Security Information and Event Management

The concept of SIEM combines Security Information Management (SIM) and Security Event Management (SEM) into a one coherent application stack with the purpose of providing a holistic and unified view into infrastructure, compliance and log management of IT environments (Piggeé Sr. 2016). In most deployments, SIEM is usually combined with a log management system to simplify the architecture of the system,

The fundamental idea of SIEM is to put collected log and event data from servers, network equipment, business applications and other systems into a single system in which it can be processed further to find deviations from the baseline or find information about an event of interest, such as a suspected data breach or unauthorized account modifications. (Piggeé Sr. 2016.)

4.1 Overview of SIEM

Technology behind SIEM and the concept of event management has existed since the late 1990s and there have been commercial SIEM systems available since the 1997. The original use case for a SIEM system was to reduce the amount of false positive detections originating from intrusion detection systems (IDS), but has only recently started to make its depute as an industry standard way to process information gathered from systems and different security solutions. (Chuvakin 2010: 2.)

A common definition for a SIEM system is that it is used to collect relevant logs, aggregate and normalize them; perform analysis either manually or automatically; present data through dashboards, reports and visualisation and execute security related workflows, such as notifications, create temporary restrictions or perform other typical security related tasks. (Chuvakin 2010: 2)

Log management is closely related to SIEM and is usually used as a basis for the system. Depending on definition and solution used, log management could also be part of the SIEM system. Log management systems are used to handle log collection, aggregation, retention of the original data and the presentation of the data in forms of searches or database queries. (Chuvakin 2010: 2.)

The difference between these two are; SIEM systems focus on the security aspect of the collected data and are only using the data which are relevant for the security of the environment, and based on that provide information which can be used to identity threats or breaches. Log management systems are collecting and managing the log files, which can be used by a wide range of applications outside of the security domain as well and in the scope of SIEM, log management tools are used to keep track of all the logs, which have been collected. Figure 1 shows the relationship between SIEM and Log Management.

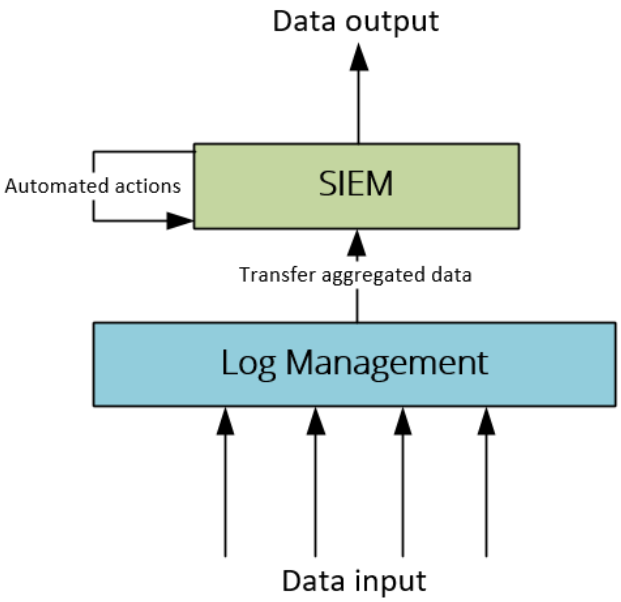

Figure 1. Relationship between SIEM and Log Management.

In most deployments, the content gathered by log management is presented to SIEM and then processed further, as demonstrated in figure 1, but some solutions do not differentiate the systems from one another. The data from log management must be processed further to be effectively used and provide significant information about the data source. (Chuvakin 2010: 5.)

The recent development done in the field of machine learning has started to take the commercial SIEM solutions to a completely new level by providing automated risk assessment and effective automated analysis of massive sets of data gathered from tens of thousands of devices resulting in billions of rows of raw data to process. The traditional

systems are focusing on signatures or manual investigation to detect intrusions and security threats, which has become nearly impossible without behaviour pattern analysis provided by machine learning. (Lunetta 2016.)

4.2 Defining Features of SIEM

The features of SIEM vary depending on system and implementation, but most deployments include the fundamental features, which are building the basis for any SIEM system, and are something most companies are looking for when evaluating a SIEM system.

The defining SIEM features are discussed below. The features described discussed are merely basic features that are covered by most commercial and open-source SIEM systems, but most of them have additional features or are focusing on a specific use case or environment.

Log and context data collection is in the core of any SIEM system and it includes data input of source data (log files, events, etc.) and context data from other sources. The context data includes additional information about the specific environment, such as information about protected users or results of vulnerability assessments. (Chuvakin 2010: 3.) Based on the context data, the SIEM system could for example create a notification whenever a protected user group, such as domain admins, is modified to detect possible unauthorized activity.

Data is collected by using a combination of agentless and agent-based methods (Chuvakin 2010: 3). Some applications are natively supporting a protocol, such as HTTP or TCP, based data input, but for some, an agent running on the data source must ship the logs to the centralized storage.

Normalization and categorization covers the processing of collected data into a format in which it can be processed by the system in a standard way. Events are also categorized into pre-defined categories, such as “File Access”, “Password Change” or “Group Modification” (Chuvakin 2010: 3), which can be further processed based on category based rules and workflows.

Notifications are used to inform security operators about events, which might be a cause for alarm and should be investigated further. In most cases, the system should be

able to produce e-mail and SMS notifications to differentiate notifications based on the severity of the event. (Chuvakin 2010: 3.)

Correlation uses algorithms, statistics, various rules and other methods to relate events with other events and events to context data. Correlation could for example relate all events which have something to do with attempts to exploit a known vulnerability (Chuvakin 2010: 3), which would make it easier to find out whether a vulnerability has been successfully exploited or not.

Visualization is utilized to create dashboards and other displays, which can be easily used by security personnel to be able to see information of interest with a quick glance without having to take a deep dive and search through the logs manually, as seen in figure 2. The dashboards can show live-data updated on a regular interval or historical data retrieved from the database. (Chuvakin 2010: 3.)

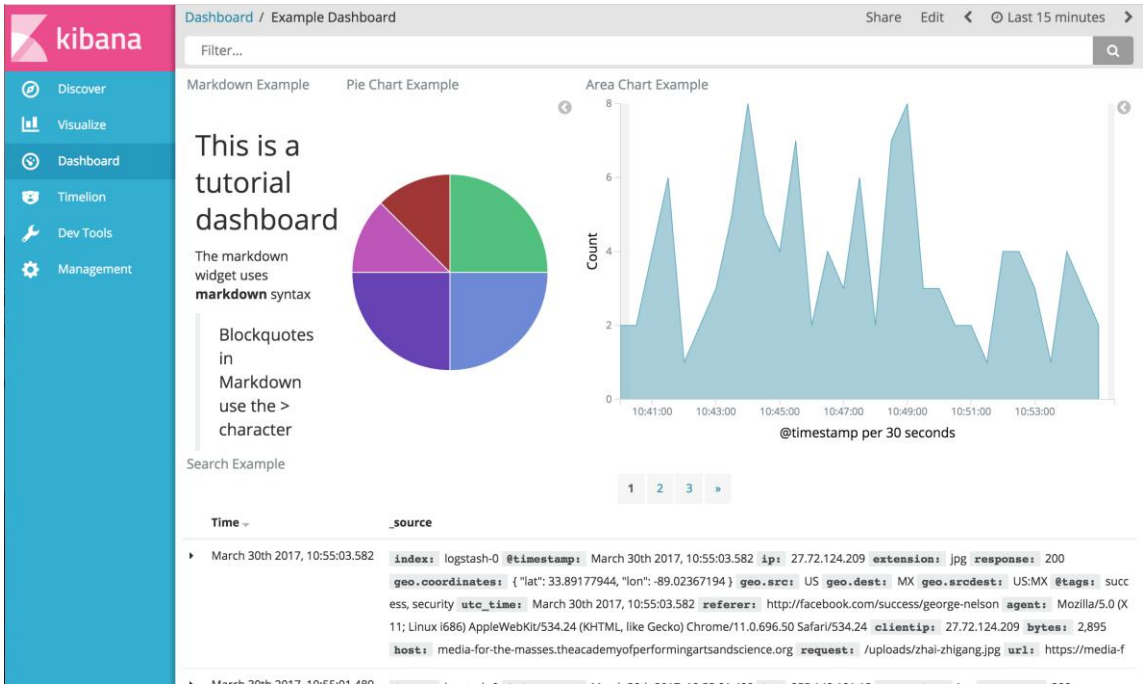

Figure 2. Picture of the Kibana tutorial dashboard.

Prioritization includes features that can be used to highlight potentially important data and suppress data, which is of lower significance. Prioritization is closely related to correlation as, for example, events that are correlated to a specific vulnerability could be prioritised higher than those that are not. (Chuvakin 2010: 3.) Events and syslog entries

usually have severity tags, which are often considered by the algorithms when prioritizing the events.

Reporting provides IT management and other personnel a simple coverage on the significant historical events and other information, which is gathered by the SIEM system (Chuvakin 2010: 3). Reports are also useful when analysing information from a longer period of time and for example, the increase in failed login attempts after a certain date might indicate an issue with operating system update applied prior to that date.

Workflows extends the capabilities of the SIEM system to perform incident management features, such as opening cases for suspicious activities or perform additional tasks either automatically or semi-automatically depending on the type of situation. (Chuvakin 2010: 3.) For example, if a certificate is used to authenticate user in a way which would be considered impossible, such as traveling from one location to another too fast, the certificate could be revoked temporarily effectively preventing it from being used for authentication purposes.

Vulnerability scanners can be used to complement the defining features of SIEM. Vulnerability scanner can be used to identity and classify security vulnerabilities found in servers, networking or other parts of the infrastructure. In a typical scenario, information from vulnerability scanners are used to identify the risks and a report is created based on the identified risks. The report could be sent to the SIEM solution as context data, which enables the system to map suspicious activities to specific vulnerabilities. (Scarfone et al., 2008: 27.)

4.3 SIEM Solutions

According to Gartner analysis (Kavanagh & Bussa 2017), the current market leaders for commercial SIEM systems, in terms of visionary and ability to execute, are IBM QRadar, Splunk Enterprise Security (ES) and LogRhythm. Micro Focus (ArcSight), McAfee (Intel) and Dell Technologies (RSA) are slightly behind the top three with their own SIEM solutions.

Open Source Security Information Management (OSSIM) developed by AlientVault is one of the few comprehensive open-source SIEM systems available, and it is mostly a

derivative of Unified Security Management™ (USM), which is a commercial SIEM system developed by the same company.

The features of the top three SIEM systems are compared in table 1 together with two open-source alternatives, OSSIM and Elastic Stack.

	IBM QRadar	Splunk ES	LogRhythm	OSSIM	Elastic Stack
Log and context data collection	Yes	Yes	Yes	Yes	Yes
Normalization	Yes	Yes	Yes	Yes	Yes
Notifications	Yes	Yes	Yes	Yes	No
Correlation	Yes	Yes	Yes	Yes	No
Prioritization	Yes	Yes	Yes	Yes	No
Visualization	Yes	Yes	Yes	No	Yes
Reporting	Yes	Yes	Yes	No	No
Workflows	Yes	Yes	Yes	Yes	No
Log Management	Yes	Yes	Yes	No	Yes
Vulnerability Assesment	Yes	Yes	Yes	Yes	No
Deployment	Software, SaaS or appliance	Software or SaaS	Software or appliance	Software	Software or SaaS
Pricing	Subscription or perpetual license	Amount of daily data	Subscription	Open-source	Open-source
Target audience	Enterprises	Regulated Industries	Midrange to Enterprise	SMB	IT Professionals

Table 1. Comparison of SIEM solutions.

As seen from the comparison in table 1, the top three commercial solutions all include the defining features of SIEM, as well as Log Management, which by definition is not part of a SIEM solution but is often included in the solution, as it builds up the basis for any SIEM solution.

From the open-source alternatives, OSSIM is a SIEM solution and as Elastic Stack focuses on log management, the systems are distinct by nature, since they are focusing on different areas and therefore cannot really be compared with one another.

4.4 SIEM in Relation to GDPR

According to article 30 of the regulation, a data controller must maintain records of the processing activities for any data, which is under the responsibility of the data controller. This means there must be a solution in place to monitor the processing activities related to personal data, whether the processed data is stored in on-premises environments or in a cloud system. (Council Regulation (EU) 2016/679 2016: 50-51.)

While most of the systems used to process the data are generating the necessary audit logs by default, there might be multiple distinct systems that are used to process or store any personal data. This means in larger environments utilizing a log management system provides the necessary tools to properly store the generated log files.

A centralized log management system and properly securing it also helps organizations to fulfil the requirements stated in the article 32, which enforces organizations to implement appropriate security measures to ensure the level of security is in line with the risks involved. (Council Regulation (EU) 2016/679 2016: 51-52.)

In some situations, the log files could be destroyed in the original source once they have been shipped forward, which in turn decreases the possibility of leaking personal information through the log files left behind at the source server.

The regulation requires that personal data must not be stored for a longer period than necessary, which means for all log files, which contain personal data, there must be a pre-defined retention period after which the data is to be deleted. The duration of the retention is dependent on the situation and the regulation does not define exactly the length of retention, but it should be long enough to ensure potential security breaches can be investigated thoroughly. (Council Regulation (EU) 2016/679 2016: 7.)

SIEM solutions could also be used to pseudonymize the log files when there is no need to include certain information, such as IP-addresses or login names, in the actual log files, which are being used to something else, such as monitoring the overall usage patterns of an information system. By utilizing event duplication, the original log files could be stored in a different index to be used in case a security breach is detected, effectively sealing the personal information to be used later for legitimate purposes without exposing them to persons who might not be authorized to process them.

As the article 33 of the regulation also enforces the data controller to notify supervisory authority in case there is a security breach which affects any personal data, there needs to be some sort of a system in place which can be used to monitor and detect possible security breaches in a timely manner. (Council Regulation (EU) 2016/679 2016: 52.)

This in practice means organizations should adapt certain practices, such as scheduled maintenance breaks to install security updates and implement intrusion detection systems to detect anomalies within the environment.

Elastic Stack was decided to be the system to be implemented at NAPA, which can be used to fulfil the technical requirements set by the regulation, as desribed in the regulation articles 30, 32 and 33.

5 Elastic Stack

Elastic Stack is a collection of open-source products, which are being developed by a Dutch company, Elastic, with an objective to collect data from any system and search, visualize and analyse the collected data. By the definition, it is a log management system, but it is also a viable alternative for small and medium size companies to commercial SIEM products, even though it does not, by itself, meet all the definitions of a SIEM system.

Elastic Stack can be expanded with a commercial extension pack, the X-Pack, which brings it closer to a full-scale SIEM solution by enabling features such as reporting, notifications and machine learning. Further customization outside of the elastic stack can be done to bring it even closer to other alternatives, for example with integrations to other products, such as OSSEC, which is an open-source Host-based Intrusion Detection System (HIDS) (Lutz 2017).

5.1 Core Components

Elastic Stack consists of open-source components which are designed to work together to create a coherent system for managing and analysis data. The components are Elasticsearch, Logstash, Kibana and Beats, and they can be used independently or as part of a custom solution. Elasticsearch, Kibana and Logstash can be expanded with plugins and there are many community-developed plugins available, which can tailor the system to meet the requirements of a specific use case.

The latest versions of Elasticsearch and Logstash can be installed on most of the modern Linux distributions, as well as Windows Server 2012 R2, but the easiest solution is to utilize a supported Linux distribution, such as Fedora or CentOS.

5.1.1 Elasticsearch

Elasticsearch is the underlying engine providing the elastic stack its capabilities of handling the collected data. It is a near real time search platform capable of storing, searching and analysing large amount of data quickly, essentially it is a noSQL database. The system is designed to be distributed, so it can be scaled with ease when a workload

becomes too high for a single node to handle. Elasticsearch is based on Apache Lucene, which is a Java library responsible for handling the document indexing as well as searching for information contained in the indices. (Elastic.co 2017a.)

All Elasticsearch installations have a cluster, which contains one or more nodes that are holding all the data and are providing ways to search the data by utilizing the Representational State Transfer (REST) Application Programming Interface (API). Elasticsearch deployments might consist of a single cluster or have multiple clusters, which are uniquely identified by a cluster name. (Elastic.co 2017a.)

Node is a single server that is part of a single cluster and its function is to store the data and participate in the indexing and search activities of the entire cluster. It is fine to have a single node cluster but most deployments have several nodes for high availability and the performance benefits a multi-node cluster provides. Elasticsearch nodes are differentiated from each other by roles, and each node can have either one, many or all the roles depending on implementation strategy and the size of the cluster. (Elastic.co 2017a.)

Elasticsearch is keeping track of documents with similar characteristics by indices. A single cluster can have multiple indices and it is recommended to create separate indices for distinct types of datasets, for example, one for personnel related data, second one for customer data and so forth. Indices can be further split to shards which can be replicated to horizontally scale the content across multiple nodes. Replicas provide high-availability in case a shard or a node fails and allows Elasticsearch to execute searches in parallel on all replicas, effectively increasing the search throughput. (Elastic.co 2017a.)

Document in the context of Elasticsearch, is a uniquely identifiable top-level JavaScript Object Notation (JSON) serialized object that contains keys and values. The key is the name of a field or property, for example “name”, and the value can for example be a string, number, boolean or another object. Elasticsearch uses mapping files to define the structure for the documents and it has a separate mapping file for each document type stored in an index. The Elasticsearch REST API is used to insert documents into the index or update, delete or retrieve the documents already stored in an index. (Elastic.co 2017a.)

For example, to store a document into Elasticsearch, a following REST API request could be used:

```
POST /indexName/typeName/
{
      “source”:             “127.0.0.1”,
      “severity”:           “5”,
      “message”:            “Authentication failed!”,
      “geolocation”: {
            “lat”:          76.0,
            “lon”:          22.0
      },
      “username”:           “admin”
}
```

The sample request above would create a document into the Elasticsearch _index field set to “indexName” and a _type field set to “typeName”. The _id field, which is used to uniquely identify the document, is generated automatically when POST request is being used. Using PUT request requires the _id field to be present in the request.

The document created by the sample above would look like this:

```
{
      “source”:             127.0.0.1,
      “severity”:           5,
      “message”:            “Authentication failed!”,
      “geolocation”: {
            “lat”:          76.0,
            “lon”:          22.0
      },
      “username”:           “admin”
}
```

The document has been stored into Elasticsearch by using a specific mapping file for the document and index. As seen, the value on the source and severity keys are mapped correctly, which makes them searchable by utilizing the correctly mapped fields. For example, to search all events with a less than 4 as the severity, the following search can be used:

```
GET /indexName/typeName/_search
{
      “query”: {
            “severity”: {
                  “lt”: 4
            }
      }
}
```

For the example document, the following mapping file was used to map the document:

```
{
      "indexName": {
            "mappings": {
                  "typeName": {
                        "properties": {
                              "source": { "type": "ip" },
                              "severity": { "type": "integer" },
                              "message": { "type": "text" },
                              "geolocation": { "type": "geo_point"},
                              "timestamp": { "type": "date"}
                        }
                  }
            }
      }
}
```

Each cluster must have a single master node that is responsible for performing all cluster-wide actions, such as tracking nodes, creating indices or allocating shards to different nodes. As the cluster health is dependent on the stability of the master node, a large cluster should have dedicated master nodes that do not take part of handling the data. Instead, the data is handled by dedicated data nodes. (Elastic.co 2017a.)

Data nodes are responsible for holding the shards containing the documents that have been indexed, and as a result, they are handling all the data related operations, such as searching and aggregations. (Elastic.co 2017a.)

Figure 3 shows an example of Elasticsearch cluster with a single master node without any associated data and three data nodes holding three primary shards with two replicas. In this scenario, the workload is split even across the data nodes as each of them holds a copy of the shard. The cluster with three primary shards and two replica shards can handle the simultaneous failure of two data nodes without interrupting the service. (Elastic.co 2017a.)

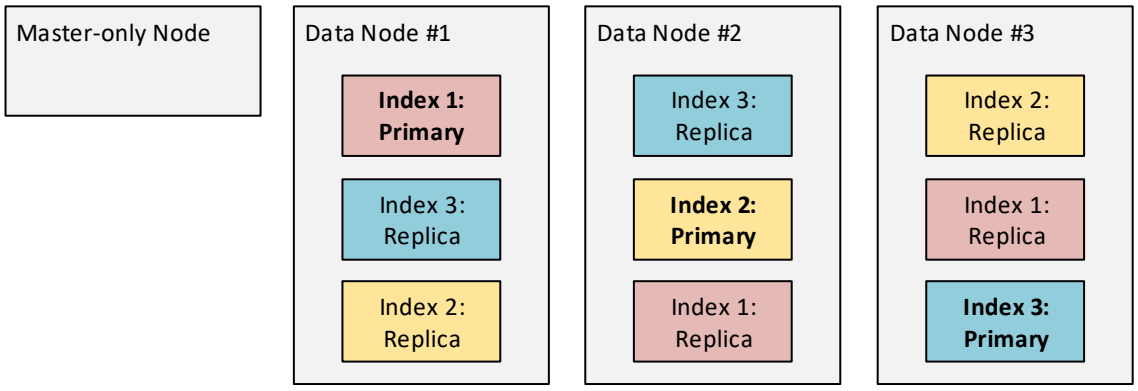

Figure 3. Example of an Elasticsearch cluster with three data nodes

In addition to master and data nodes, Elasticsearch cluster can also include Ingest nodes, which are used to pre-process documents before the indexing takes place. The node intercepts index requests received by the cluster, performs transformations and passes the documents back to the index. Transformations are done based on processors defined in pipelines and each processor modifies the data in some way. For example, a processor could be created to add additional information to a document whenever it contains a certain combination of keywords. (Elastic.co 2017a.)

5.1.2 Logstash

Logstash is a data collection engine, which can in real time, unify the data from multiple sources and normalize it into a destination system. In Elastic Stack, the information is send to Elasticsearch for further analysis but the information can be stashed in other locations as well, such as MongoDB or Amazon S3 bucket. (Elastic.co 2017b.)

One of the fundamental ideas behind Logstash is the ability to clean, transform and enrich any of the events before they are being sent to downstream for analysis or visualization. Logstash can be expanded with plugins and has out-of-box support for many popular log file formats, such as syslog or Apache log files. (Elastic.co 2017b.)

The event processing functionality in Logstash operates in three stages as seen in figure 4. The first stage is input, which generates events. The second stage is filtering, which modifies the original events by creating additional fields or combining corresponding events from multiple sources into a single event. It can also mutate the events or drop them completely, which could be useful to for example, filter out debug events. The final phase is the output, which passes the event to a downstream system. (Elastic.co 2017b.)

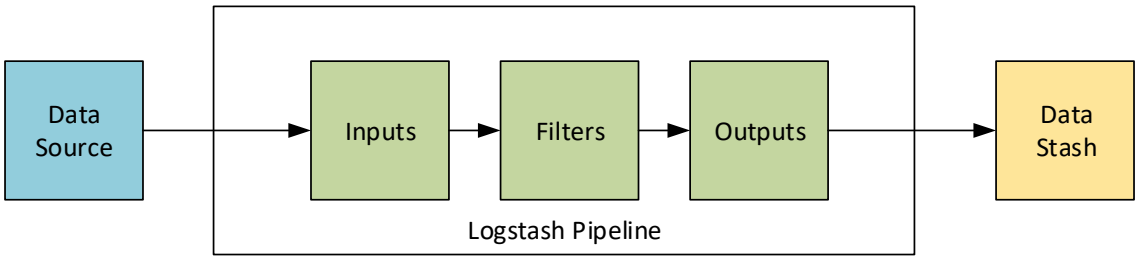

Figure 4. Logstash pipeline stages (Elastic.co 2017b).

Logstash runs by default only a single pipeline, but the configuration can be changed to

allow multiple pipelines to exist in the same process. Utilizing multiple pipelines is useful whenever the configuration has distinct event flows that do not share same inputs, filters or outputs and they are separated from each other with tags. (Elastic.co 2017b.)

With Grok filter plugin, Logstash can parse unstructured log data into structured format, which can be queried and processed by a downstream system. Under the hood, Grok uses a regular expression to parse the messages to create additional fields to events by separating items from the original log message. IP and URIPATHPARAM are IDs for regex patterns, which are defined in the Grok pattern file, and they are used to parse an Internet Protocol (IP) address and a Uniform Resource Locator (URL) path from incoming string. (Elastic.co 2017b.) For example, the plugin would be able to parse the partial and fictional HTTP request log:

```
77.85.101.44 GET /index.php
```

The HTTP request can be parsed with Grok by using the following filter:

```
%{IP:source} %{WORD:method} %{URIPATHPARAM:request}
```

This filter would add three extra fields to the event that can be easily queried by the downstream system. Without filtering, the event would contain just a single line containing all the information, making it more difficult to query.

```
source: 77.85.101.44
method: GET
request: /index.php
```

With this information alone, it would be possible to create a visualization showing the most popular file requests on a particular web server. With additional enrichment, such as geo coordinates, it would be possible visually to see from which countries the requests originate and create a heat map to quickly visualize from which areas people connect to the server.

5.1.3 Kibana

Kibana is a platform designed to be used with Elasticsearch with a purpose to help the users to analyse, browse and visualize data stored in Elasticsearch indices. It has a simple browser based interface which can be used to create dynamic dashboards and

search data directly from the indices using either Lucene query syntax or JSON-based Elasticsearch syntax, both which are enabled by default. Version 6.0 also includes a new experimental query language, Kuery, which is designed to be especially used with Kibana. (Elastic.co 2017c.)

The Discover page, as seen in Figure 5, provides an interactive way to explore the data stored in Elasticsearch by displaying all the documents, which match to the specified query (Elastic.co 2017c). For example, to search all the http requests with a response code 404 (not found), the following query based on the Lucene syntax could be used:

```
type: http AND http.response.code: 404
```

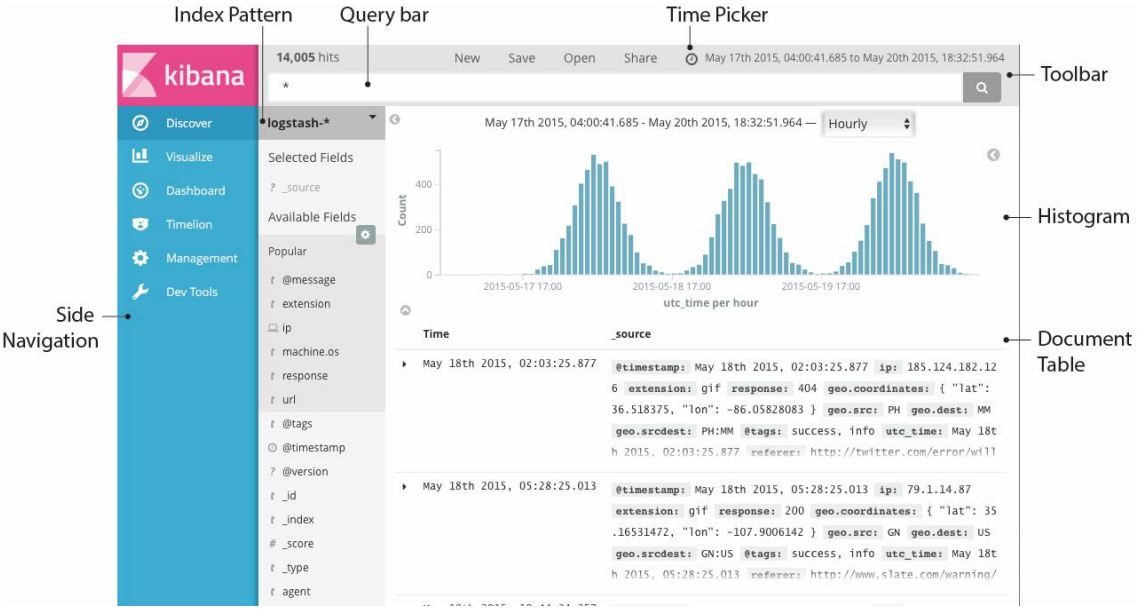

Figure 5. Kibana Discover page (Elastic.co 2017c).

For comparison, an equivalent query in Structured Query Language (SQL) would be:

```
SELECT * FROM indexName WHERE type = “http” AND http.response.code = 404;
```

The search syntax follows the Lucene syntax, which means you can search from specific fields by separating the field from the value by a colon (key:value) and use logical operations to combine multiple fields in the query. The information returned could be used to identify if the website has been under an attack utilizing URL fuzzing techniques to find vulnerabilities or hidden data directories.

Kibana also has visualization capabilities which are often used as a part of dashboard. Dashboards are a collection of visualizations, such as aggregations and charts, which have been created on information received from Elasticsearch queries. (Elastic.co 2017c.) Dashboards are often focusing on a specific area of interest, such as http server statistics or syslog redirected from various networking equipment.

Timelion can be used to combine two or more independent data sources together to view them at the same time in a same visualization, which enables the possibility to for example compare Central Processing Unit (CPU) usage from last hour to the weekly average (Elastic.co 2017c).

5.1.4 Beats Platform

The Beats Platform was introduced in version 5.0 of Elastic Stack and it is the latest addition to the Elastic Stack. The Beats Platform consists of various Beats, which are open-source data shippers designed to be installed as agents on servers to send operational data either directly to Elasticsearch or via Logstash. (Elastic.co 2017d.)

The beats are installed separately and each beat is functioning as an individual beat and usually only fulfils a single purpose. There are four beats which are managed by Elastic; Packetbeat, Filebeat, Metricbeat and Winlogbeat. In addition to these, there are many community beats to fill a specific use case, such as wmibeat, which can run Windows Management Instrumentation (WMI) queries to retrieve metrics from Windows based operating systems. (Elastic.co 2017d.)

The beats are built on top of the libbeat library, which contains the necessary components for a beat to function consistently and properly communicate with Elasticsearch and Logstash. The libbeat library provides a straightforward way to create custom beats to meet specific use cases, which removes the need to have a single agent responsible for all the different use cases organizations might have. (Elastic.co 2017d.)

For beats, the load-balancing functionality is built-in to the configuration file of the beat, but for other input methods an external load-balancing tool, such as HAProxy, or a message queuing system, such as Kafka, could be used instead.

5.1.5 X-Pack

The X-Pack is a commercial subscription-based addition to the Elastic Stack, which bundles together several components to enable new capabilities, such as additional security measures, status monitoring, alerting, machine learning and graph analytics. In addition to open-source licenses, basic and no-license levels, there are three paid subscription levels; gold, platinum and enterprise, each one providing additional features with an increased subscription cost. (Elastic.co 2017e.)

The Security functionality of X-Pack contains features that are essential to securely operate the elastic stack and ensure compliance with various regulations, such as the GDPR for example. The log files stored in Elasticsearch are likely to contain sensitive information, which is the reason why the system must have a comprehensive auditing functionality, which the basic installation of elastic stack is lacking. The security functionality also includes role-based access control, field- and document-level security settings and encrypted communication.

Since the Elastic Stack is open-source, there are some alternatives to the features contained in X-Pack, such as Search Guard, which provides similar functionality as the X-Pack security feature does, but with a smaller price (Floragunn 2017). The community edition includes access controls based on document type and index, which can be used to restrict access to potentially sensitive data by placing them in to a separate index.

5.2 Availability and Performance

All the components of Elastic Stack have been designed to be horizontally scaled to enable high availability and resource optimization depending on the usage scenario. Kibana and Logstash are operating independently and do not have native tools to enable clustering or high-availability configuration.

Figure 6 shows an example of a redundant Elastic Stack architecture with two Logstash instances, ten Elasticsearch nodes and two Kibana instances. The single coordinate-only mode is responsible for load-balancing the requests across the cluster and in case it would go down, any of the other nodes would start to coordinate the requests sent to the cluster.

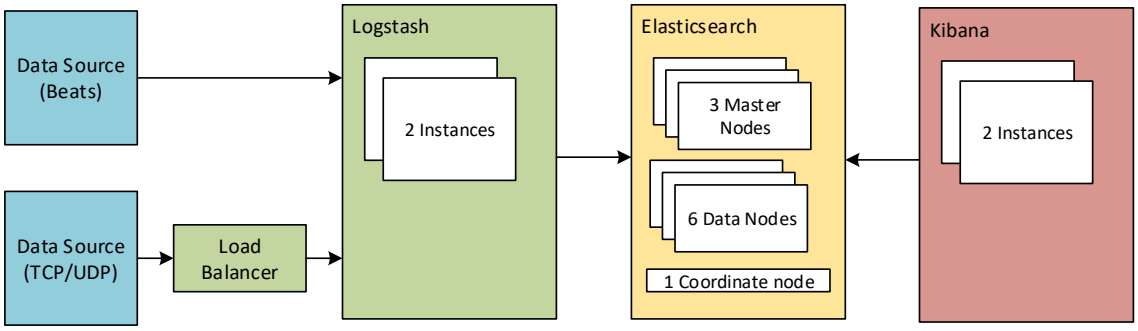

Figure 6. Example of redundant Elastic Stack architecture

To achieve high-availability and to ensure the Elasticsearch cluster does not enter a split-brain situation, the cluster should have at least three master-eligible nodes when the minimum_master_nodes configuration parameter is set to 2. The parameter sets the value of least number of master-eligible nodes, which need to be available to start the election process for a new master. If there are only two master-eligible nodes and the parameter is set to 1, a network connectivity problem between the nodes will cause each node to elect themselves to be the new master node which would cause a split-brain situation to occur. (Elastic.co 2017a.)

Coordinating-only node is an Elasticsearch node in which all the roles (master, data & ingest) are set to false. The coordinating-only node can be used to perform load-balancing across the Elasticsearch cluster and the node will process incoming HTTP requests and redirect them to other nodes accordingly. The coordinating-only node is mostly used to balance the request coming from Logstash or Kibana instances. (Elastic.co 2017a.) The coordinating-only node can also be installed on the Kibana instance itself without needing a separate instance to run the coordinating-only node (Elastic.co 2017c).

Kibana does not perform any resource intensive actions by itself and it does not require extensive amount of computing resources. Multiple Kibana instances are usually only needed for redundancy or load-balancing when there are many simultaneous users connecting to the system.

With the default configuration, Logstash utilizes memory-bound queues between the pipeline stages, but to increase resiliency and allow the usage of larger buffer, persistent queues can be used instead. With persistent queues enabled, the buffer stages are stored on disk instead of utilizing the memory-bound buffer, which provides additional safeguard against data loss in case a Logstash node would fail. Using persistent queues

will affect the input/output performance of the host device and therefore, it is recommended to use solid state drive backed storage when persistent queues are in use. (Elastic.co 2017b.)

The performance of Logstash instance is largely dependent on the quality of the source data and filter configuration, since Logstash spends most of its processing time on applying the filters and transforming the data. When possible, a structured file format, such as JSON or Extensible Markup Language (XML), should be used instead of relying solely on Grok to parse data input, as the performance of grok degrades considerably whenever a regular expression fails to match a string or it must rely on negative matches to produce a filtered output. (Atanasov 2017.)

5.3 Elastic Stack as a Full-scale SIEM?

Turning Elastic Stack into a full-scale SIEM solution will require the usage of X-Pack, or some of the alternatives, and integration of additional 3rd party components to cover vulnerability assessment and detection of intrusions. Correlation and prioritization can be achieved by creating sufficient pre-processing pipelines in Logstash, for example if an attempt to exploit a vulnerability is recorded from a specific source IP, all incoming documents which contain the same source IP would be marked as important and linked to the exploit attempt.

With Elastic Stack, the solution can be tailored to meet the needs of a specific environment. For example, a traditional corporate environment with managed systems has completely different needs than a modern start-up would have. Both environments would benefit from a SIEM solution, but their optimal implementations differ from each other quite drastically.

Therefore, the process of turning Elastic Stack into a SIEM starts by creating a list of requirements that the solution should fulfil. For example, a company involved with payment card processing must comply with Payment Card Industry Data Security Standard (PCI-DSS), which clearly defines concrete levels of security the company must achieve (PCI Security Standards Council, LLC 2016: 8). This means whatever system is being used, must be capable of fulfilling the requirements set by PCI-DSS.

One of the key considerations when planning to use Elastic Stack as a SIEM is to think about the data sources and potential points of intrusions, and how to ensure all the necessary information is available in the Elasticsearch. It is likely many of the applications will require a custom Beat to be created to efficiently gather the necessary data or enrich it before shipping it forward.

Correlation is also something which must be considered in detail. A simple event, such as establishing a virtual private network (VPN) connection, might generate related events in various systems, such as firewall and the identity provider, which authorizes the VPN connection. If the data is not available or searchable, an intrusion might become unnoticeable by blending in with legitimate events. Most commercial solutions include built-in correlation rules for various systems and have more advanced methods to correlate the data out-of-box, making them easier to configure.

Using Elastic Stack as a SIEM is more labour intensive than deploying a commercial SIEM solution with everything built-in, but the process of manually defining everything and planning the architecture provides additional insight on how things work and as a result the end-product might be more beneficial than a commercial SIEM could be. The Elastic Stack can also be used as a search engine and database replacement to cover additional use cases which differ from a traditional SIEM solution.

5.4 Considerations for the Implementation

When planning for Elastic Stack deployment, there are a few things which need to be considered before the implementation process is started. Disk usage on Elasticsearch nodes can become a problem unless it is accounted for during the initial stages of the implementation. In addition to disk usage, inefficient memory allocation might cause performance issues or waste considerable amount of memory in the process.

By default, Elasticsearch does not allocate additional shards if more than 85% disk space is being used and this is something that needs to be accounted for, especially on deployments with high-availability requirements Elastic (2017a). Large deployments usually have multiple nodes and are utilizing replica shards and since replica shard contains the exact same information as the primary shard, they are similar in size which effectively doubles the raw capacity needed across the cluster.

Disk space needed in a deployment with multiple nodes and replica shards can be calculated with an equation:

$$N_{capacity} = \frac{\text{retention} * data_{in}}{0{,}85} * (R_{count} + 1) \; / \; (N_{count} - F_{tolerance}) \quad (1)$$

where $N_{capacity}$ indicates the total capacity needed for a single node, $data_{in}$ the incoming amount of daily data, R_{count} the count of replica shards, $F_{tolerance}$ the tolerance for node failures and N_{count} the total number of nodes.

As a result, it is possible to determine that in a three-node configuration with two replica shards and a fault tolerance of a single node, each node would require 5,55 terabytes of storage, in an implementation in which two different indices receive 35 gigabytes data in a week with thirty- and sixty-day retentions.

According to Elasticsearch: The Definitive Guide (Gormley & Tong 2015), most deployments do not require extensive CPU resources and other resources, such as memory and storage, affect the performance more than the CPU.

Memory allocation is also a crucial factor to consider when planning for Elasticsearch deployment, since Java is using compressed ordinary object pointers (OOP), to circumvent overhead when using 64-bit pointers. By using compressed OOPs the pointers reference object offsets instead of exact byte locations in memory, effectively allowing the 32-bit pointer to reference up to four billion objects instead of bytes. In practice, this means a heap can be up to 32 gigabytes in size while still utilizing 32-bit pointers, which decreases wasted space and bandwidth required to transfer data between memory and system caches. (Gormley & Tong 2015.)

Because of the trick, the recommended heap size is slightly under 32 gigabytes to allow Java to utilize compressed OOPs instead of having to rely on 64-bit pointers, which would effectively waste memory due to increased overhead. It would take additional 8-18 gigabytes of allocated memory to overcome the overhead, considerably increasing the memory consumption without any significant performance improvements. (Gormley & Tong 2015.)

For overall memory allocation, the recommended practise is to allocate half of available memory to Elasticsearch heap and allow Lucene to utilize rest of the available memory.

The memory assigned to Elasticsearch heap should be as close to the amount required with minimal overhead, as it will offer a noticeably higher performance by making the garbage collection faster. (Gormley & Tong 2015.)

6 Elastic Stack Implementation

The implementation of the Elastic Stack for NAPA will be quite lightweight in the beginning but the plan is to expand it in the future, once the basic things are up and running. The implementation is done on a single physical server and is hosting all the necessary components of Elastic Stack.

Since the implementation of any SIEM system is a lengthy process, it was decided to leave systems that are not in the centre of the regulation out of this thesis, as each system and server should be modelled accurately to ensure the most optimal outcome. Therefore, it was concluded to approach this chapter in a more generic way.

The content of this chapter is based, unless otherwise stated, on the experimentation of the author and knowledge received while researching Elastic Stack during the previous chapter.

6.1 Elastic Stack Preparations

Before the project was started, a list of services and systems from which log files should be gathered from was created and each service was placed in either SIEM index or Log index. SIEM index is an Elasticsearch index for security related events, such as failed user authentications or anything related to the regulation. The log index is an index in which information which could be useful in case a breach occurs, but have no value on their own, such as information about Dynamic Host Configuration Protocol (DHCP) leases.

A virtual test environment was installed before the start of the project to estimate how much disk space and computing resources are needed for the implementation consisting of all the systems chosen for the initial implementation. The retention period used in the estimation was forty-five days and a simple equation derived from the equation introduced in chapter 5.4 was used to estimate the disk usage.

$$N_{capacity} = (45 * data_{in}) * 3 \quad (2)$$

where $N_{capacity}$ is the total space required for the indices and $data_{in}$ is the average of incoming daily data. The result was multiplied by three, as the test environment was receiving data from roughly one third of the systems to be included in the actual implementation.

As mentioned in chapter 5.4, Elasticsearch does not need an extensive amount of CPU resources, and this was also something that was observed when the virtualized environment was setup the first time. The CPU load constantly remained under 5% during normal operation and even under extreme stress did not exceed 15%, with four virtual CPUs on a hypervisor with Intel Xeon E5-2670 CPU.

The server responsible for hosting Elasticsearch has 32 gigabytes of memory available, which means the compressed OOPs trick mentioned in chapter 5.6 does not need to be considered during this implementation. Following the recommended memory allocation practices, half of the available memory should be assigned to Elasticsearch. However, as the same server will be responsible for hosting Logstash and Kibana in addition to Elasticsearch, the memory allocation should account for their needs as well.

Since Logstash is running on Java Virtual Machine (JVM), it will consume all the memory it is being allocated at start-up and therefore, the actual memory requirements are hard to estimate without experimentation, as the memory usage is dependent on the size of the implementation. For this implementation, allocating sixteen gigabytes to Elasticsearch leaving the others to run with the default options provides adequate performance for all operations. The memory allocated to Logstash might have to be increased as the usage of the system is expanded.

6.2 Preparing the IT Environment at NAPA for the Elastic Stack

To securely operate any SIEM or log management solution, the surrounding environment should be prepared properly to function with the solution that is being implemented, since compromised or faulty logs will provide no real value and might even endanger the integrity of all the other log files making them unusable in case they are needed in a legal case.

As seen in chapter 5.1.2, Logstash has a native support for syslog input and it can transform syslog messages to events over the network connection, if the syslog format on the

source device complies with Request for Comments (RFC) 3164, Logstash can directly parse the message without processing it any further.

The problem with the default remote syslog implementation is the lack of encryption in the RFC 3164 standard (Lonvick 2001), which means remote syslog messages sent from devices without the possibility to run a separate log collector, such as network appliances, are sent unencrypted and therefore are vulnerable to man-in-the-middle attacks. Modern network appliances have a support for Transport Layer Security (TLS) encryption as well and to use encryption with Logstash, a custom input filter has to be created for the remote syslog originating from these devices.

To mitigate this vulnerability, the syslog messages from network appliances should be sent over an encrypted channel or a separate out-of-bound connection dedicated for management traffic. If the data is not encrypted, there is a possibility it can be modified, replayed or sniffed in transit, making it impossible to completely mitigate the vulnerability until proper encryption methods are applied.

For servers that are supporting separate log collectors the recommended method is to use them instead. For example, Filebeat can ship the syslog messages stored on a Linux server, as well as other log files that might reside in various locations. Filebeat and other beats in general are supporting mutual authentication which should always be implemented to ensure the Logstash server receives encrypted traffic only from trusted clients and that the clients do not send the files to an untrusted server.

Managing the beat agents on Linux-based systems with a package manager is straightforward as well, since the packages are available in the Elastic repository from which they can be installed directly, but for Windows systems, the agent management is a little bit more complex, since there are no native tools to manage the installations. Regardless, the ability to install the agent is not usually enough since it will also need a configuration to function properly.

Many of the typical use cases, such as Windows events and generic log files, can be fulfilled with the default Beats, but some unique applications will require a custom Beat to be configured to ship information to Elastic Stack. For example, M-Files could benefit from a custom Beat to enrich the log files with the necessary information, to differentiate the files which are related to the regulation from other files.

M-Files saves the log files in XML format, which contains information about which file has been accessed, the performed action and the person who accessed the file. However, the log file does not include any of the file attributes, such as information whether the file contains personal data or not and as such provides no information whether the file in question is under the regulation or not.

To successfully monitor the usage of files that are under the regulation, the information must be visible in Elasticsearch, since the naming practise for the files varies and it would be unreasonable to assume all the files which contain personal data are named in a similar fashion. Therefore, a custom solution is needed to enrich the log files before transferring them to Elasticsearch.

6.2.1 Configuration Management

Software configuration management tools, such as Puppet or Chef, can be used to create a configuration baseline of a device that can be used to ensure a specific configuration is in place on the managed device. Configuration management tools can also be used to distribute new settings, such as certificates or new configuration files to the managed device, which helps to ensure the configuration files are setup accordingly.

A configuration baseline could contain a list of required software and their respective configuration files, and for example, the beats agents could be managed with the configuration management system to ensure any new configuration parameters are automatically distributed to all the agents. On Linux systems, a native package manager, such as yum or apt, can be used to install a specific beat and after installation, the configuration file could be fetched from the configuration repository.

On Windows systems, a third-party package manager, such as Chocolatey, could be used since the operating systems do not have a native package manager. Puppet can also manage Linux systems, making it a suitable candidate to consider when a mixed operating system environment is being configured to function with the Elastic Stack. (Reynolds 2016.)

As mentioned in chapter 5.2, there are no built-in tools to manage the Kibana configuration file to ensure the instances have same configuration, but this can be achieved by utilizing a configuration manager to keep the kibana.yml file the same on all instances.

Kibana dashboards are saved in Elasticsearch under a Kibana index and as a result, they do not need to be replicated from one instance to another.

In a high-availability deployment, the Logstash instances are functioning as individual instances and are not dependent on each other. As stated in chapter 5.2, there is no built-in clustering functionality, which means the configuration files across the instances must be maintained manually or by using a configuration management tool. There is currently an open issue in Logstash GitHub repository regarding the support for native clustering, but the issue state is currently open and there are no plans for implementation (Rao 2018).

In addition to the configuration files, the instances responsible for hosting the Elastic Stack components need certificates from a trusted Certificate Authority (CA), in order to uniquely identify all the clients from one another, which enables mutual authentication between Elastic Stack components.

6.3 Elastic Stack Configuration and Security

With the default configuration, Elasticsearch runs in a development mode and will continue to do so until network settings have been properly configured. Elasticsearch will require more resources than users are allowed by default, which means certain settings must be modified before going into production. Elasticsearch will execute bootstrap checks during start-up to check for configuration problems and whether the required parameters are set correctly. When starting Elasticsearch in development mode any failures during the bootstrap checks will cause a warning to appear in log file, but in production mode, Elasticsearch will refuse to run. (Elastic.co 2017a.)

The settings which need to be covered before going into production are defined in the Elasticsearch manual as:

> Set JVM heap size, disable swapping, increase file descriptors, ensure sufficient virtual memory and ensure sufficient threads (Elastic.co 2017a).

Swapping is also something that should be disabled on Logstash instances as well, as it will degrade the system performance if parts of the JVM heap are swapped out to disk. If this happens, it causes the garbage collection to last considerably longer than it should,

which causes severe performance issues with Elasticsearch and Logstash processes, and it might even cause a Elasticsearch node to disconnect itself from the cluster. (Elastic.co 2017a.)

When installing Elasticsearch from .rpm or .deb package, file descriptors, virtual memory and thread limits are setup accordingly during the installation. However, if the system uses systemd, the limits need to be manually defined in the Elasticsearch systemd configuration file.

The pipeline configuration file for Logstash has a separate section for each plugin in the event processing pipeline and in most cases, the configuration file consists of input, filter and output sections. Upon start-up, Logstash combines all the .conf files contained in the default configuration directory to a one single configuration file and therefore, depending on personal preference, the configurations can be maintained either in separate files or in a single configuration file.

As mentioned in chapter 5.1.2, Logstash runs by default only a single pipeline and since the pipeline does not accept any new input before the current batch is processed, there is a risk of congestion if for example the output destination is unreachable. With a single pipeline, the processing of all events would stop until the destination becomes available once again.

Logstash, Beats and Kibana all have TLS support available in the open-source product, but Elasticsearch will either require the commercial X-Pack (Carey 2017), open-source alternative, such as Search Guard, a self-made proxy or a load-balancer between the Elasticsearch nodes. The communication between the Elasticsearch nodes must be encrypted and mutual authentication enabled to prevent data from being modified or sniffed in transit and to ensure both parties are who they claim to be.

The open-source version of Elastic Stack also lacks the possibility to enable document, index or field-level access controls, which can be problematic in a deployment with multiple user groups with different security levels (Elastic.co 2017f). The documents stored in Elasticsearch might contain privileged information that is covered by the regulation, such as usernames associated with IP addresses. User authentication and auditing are also among the essential security features that are provided by the X-Pack or another alternative.

The problem with using alternatives, open-source or commercial, to enable essential security features is the lack of official support from Elastic, which might expose the environment to various vulnerabilities or exploits. It is also reasonable to assume the alternatives are not always supporting the latest versions of Elastic Stack components, making it more difficult to keep up with the latest features and security updates.

6.4 M-Files Event Log Processing

Appendix 1 shows the source code written in C# of the M-Files log extractor prototype, which extracts the event logs from the system in XML format. The extractor enriches the events by adding an additional field to indicate the relation of the document to the regulation and once the event logs have been enriched, Filebeat ships them to Logstash.

Alternatively, a completely custom beat could have been created, but since the M-Files REST API only supports user-level operations, a separate extractor had to be written by using the .NET API (M-Files 2017). This meant the custom beat would have shared most of the functionality with Filebeat, apart from executing the extractor itself, which in this case means creating a custom solution is not worthwhile, as it does not add enough additional value at this time.

One of the benefits of having a single program to handle the extracting and Logstash shipping would be the additional security from not having to rely on an intermediate storage from which Filebeat fetches the content. The usage of an intermediate storage enables a malicious user to plant specially crafted files at the directory to disrupt the service or exploit any vulnerabilities in the Logstash filters.

The extractor runs as a scheduled task and Filebeat monitors the destination directory. Whenever there are changes in the files or new ones are added to the directory, a task is triggered which will send the latest data to the specified destination. Since the XML files have events that consists of multiple lines, the Filebeat should be configured to properly parse the events.

Below is an example of a single event (logout):

```
<?xml version=”1.0”?>
<root>
      <event>
            <id>2000</id>
            <type id=”Logout”>Logout</type>
```

```
            <category id=”5”>System</category>
            <timestamp>2018-01-14 14:12:22.538000000</timestamp>
            <createdbyuser loginaccount=”hostname\username”></causedbyuser>
            <data/>
      </event>
</root>
```

As seen from the example above, the actual events are enclosed within event-tags, which means the Filebeat must be configured to utilize a multiline functionality. By default, Filebeat reads a single line as a single event, which would mean the events would become malformed and therefor unusable. To remediate this, multiline support must be enabled and a regex pattern ‘<event’ with negate set to true, can be used to separate all events into their own events.

The Logstash pipeline configuration file, as seen in Appendix 2, is responsible for transforming the event logs originating from M-Files into a format in which it can be properly searched in Elasticsearch. Logstash pipelines are functioning in a linear manner, meaning the configuration file is executed from top to bottom. The filter section has conditional separation, which is based on an additional field that has been added to the event by Filebeat.

The original event is parsed into separate fields by utilizing the XPath setting in the xml filter plugin. XPath is a query language designed to flexibly point to various parts of XML documents, making it an efficient tool to query for a data stored in a XML file (Mozilla 2018). By pointing the XPath to event node, it will extract whatever is contained within the event node, or event tags, to a new field to be used later.

The values of interest contained within the document will be declaratively retrieved from the XML document, which works fine for relatively simple documents. For complex XML documents, it would be best to use alternative methods, such as simplifying the document at the source or converting it to JSON format. Documents in JSON format can be expanded directly to a data structure within the Logstash event, making them more robust when relying on automatic field creations.

When extracting values by using XPath, the values are by default stored as an array even if there is only a single value. The XML filter has a setting to prevent single values being stored as an array, but it does not function properly with XPath, as described in

Logstash-plugins GitHub repository (Ondas 2017). As a workaround, the extracted values are converted to text by replacing the field value with the value stored in the first index of the array.

In the later stages, the default timestamp is replaced with the actual timestamp contained within the event log. The default timestamp takes the timestamp at the time when the document arrives as input to Logstash. This would make the system inaccurate when correlating events with one another, as depending on the extract interval, there could be minutes of difference between the default timestamp and the actual timestamp.

Once the document has traversed through the Logstash pipeline, it is shipped to Elasticsearch to be indexed based on the mapping file. Elasticsearch can automatically recognize certain field types and create a mapping file based on that, but the file should be generated in advance to avoid the need of re-indexing all the data in case there are problems with the automatic field mappings.

Figure 7 displays a partial table of a document visible in Kibana, which contains a single event within the M-Files event log. The mappings have been done in a way that event, object and version ID's are stored as integers, making the fields available for numerical operations.

#	mf.eventid	3,155
t	mf.eventtype	File downloaded
t	mf.gdpr	Not defined
t	mf.loginaccount	DESKTOP-P8H0K6V\forsb
#	mf.objectid	486
#	mf.objectversion	1
t	mf.title	Non-disclosure Agreement - A&A Consulting (AEC) (6/2011).TIF
t	mf.vaultguid	{C840BE1A-5B47-4AC0-8EF7-835C166C8E24}

Figure 7. Kibana document table

As seen in Figure 7, the mf.title field could potentially contain personal information under the regulation, as the field contains the name of the document stored in M-Files, therefore a field level security constrains should be placed on the field in case the X-Pack is being used. However, the information is not relevant from a SIEM point of view, which means the field could be deleted entirely. The object ID and the information about the regulation

relation are enough to satisfy the relevant use cases, since the object ID can be used to search for the file from within the M-Files if necessary.

Since almost all the documents that contain personal information related to employees are stored in M-Files, it is possible to monitor the processing activities of the personal data, by utilizing the log extracting tool together with Elastic Stack. Additional fields to indicate to which employee the processed file is related to should be extracted as well. This is to enable employee field in search queries, which allows to locate all processing activities related to a specific employee, if the employee requests to see the information.

7 Conclusions

The technical impact of the regulation at NAPA ended up being less than originally expected and therefore, the influence the regulation had on the thesis was reduced considerably. The regulation provides a high-level picture of the technical measures that organizations need to take, but unlike PCI-DSS, it does not provide concrete levels organisations need to be at in order to be compliant with the regulation.

This changed the tone of the thesis to focus more on Elastic Stack and its implementation, instead of focusing on the regulation and implementation considerations related to the regulation.

The current implementation of the Elastic Stack at NAPA fulfils the technical requirements defined in articles 30, 32 and 33 of the regulation. The system can also be further developed to improve the security of the IT environment by implementing new processing pipelines in Logstash and creating additional integrations to other security systems, such as the intrusion detection system and client anti-virus program.

In order to be fully compliant with the regulation, additional process and policy related changes are needed at NAPA. The Elastic Stack implementation must also be made redundant and additional security measures must be implemented to ensure the integrity of the log files. Dashboards, visualizations and notifications must also be implemented, as simply gathering the data will not enable the IT team to detect security breaches or unauthorized usage of privileged information.

The process of mapping the environment for Elastic Stack turned out to provide valuable insights on the overall functionality of the IT environment, which can be seen as a major advantage of choosing the more laborious option, instead of relying on a commercial SIEM solution, which would have been easier to implement.

Based on this research regarding SIEM solutions, Elastic Stack can be a viable alternative to commercial SIEM solutions for small and medium size businesses, but in addition to the costs associated with X-Pack, the laborious implementation process must be considered when deciding which solution to use.

References

Council Regulation (EU) 2016/679 (2016). EUR-Lex - 32016R0679 – EN. [online] Available at: http://data.europa.eu/eli/reg/2016/679/oj [Accessed 12 Sep. 2017].

eugdpr.org (2016). How did we get here? [online] Available at: http://eugdpr.org/how-did-we-get-here-.html [Accessed 12 Sep. 2017].

Georges, C. (2017). What You Need to Know about GDPR. [online] Available at: http://corporatecomplianceinsights.com/need-know-gdpr/ [Accessed 5 Sep. 2017].

Information Commissioner's Office (UK) (2017). Overview of the General Data Protection Regulation (GDPR). [online] Available at: https://ico.org.uk/for-organisations/data-protection-reform/overview-of-the-gdpr/ [Accessed 5 Sep. 2017]

Gabel, D. & Hickman, T. (2016). Unlocking the EU General Data Protection Regulation: A practical handbook on the EU's new data protection law. [online] Available at: https://www.whitecase.com/publications/article/unlocking-eu-general-data-protection-regulation-practical-handbook-eus-new-data [Accessed 5 Sep. 2017].

Lång, J., Haavikko, T. & Päivinen, K. (2017). The National Implementation of the GDPR in Finland Takes the First Step. [online] Available at: http://dittmar.fi/insight/articles/di-data-protection-alert-21-june-2017 [Accessed 10 Sep. 2017].

Loyens & Loeff (2017). GDPR - Sanctions for non-compliance. [online] Available at: https://www.loyensloeff.com/en-us/news-events/news/gdpr-sanctions-for-non-compliance [Accessed 10 Sep. 2017].

i-SCOOP (2017). Personal data protection: data subject, personal data and identifiers explained [online] Available at: https://www.i-scoop.eu/gdpr/gdpr-personal-data-identifiers-pseudonymous-information [Accessed 2 Feb. 2018]

Gemalto (2017). EU Compliance: General Data Protection Regulation (GDPR). [online] Available at: https://safenet.gemalto.com/data-protection/data-compliance/european-union-eu-compliance/ [Accessed 19 Nov. 2017].

NAPA (2017). About NAPA. [online]: Available at: https://www.napa.fi/About-NAPA [Accessed 19 Nov. 2017]

Piggeé Sr., J. (2016). What is a SIEM? [online] Available at: https://www.tripwire.com/state-of-security/incident-detection/log-management-siem/what-is-a-siem/ [Accessed 12 Sep. 2017].

Chuvakin, A. (2010). The Complete Guide to Log and Event Management. [online] Available at: https://www.novell.com/docrep/documents/9x1wixnqhd/Log_Event_Mgmt_WP_DrAntonChuvakin_March2010_Single_en.pdf [Accessed 30 Sep. 2017].

Lunetta, L. (2016). Machine learning combined with behavioral analytics can make big impact on security. [online] Available at: http://thirdcertainty.com/guest-essays/machine-learning-combined-with-behavioral-analytics-can-make-big-impact-on-security/ [Accessed 30 Sep. 2017].

Scarfone, K., Souppaya, M., Cody, A., Orebaugh, A. (2008). Technical Guide to Information Security Testing and Assessment. [Online] Available at: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf [Accessed 2 Feb. 2018]

Kavanagh, M., K. & Bussa, T. (2017). Magic Quadrant for Security Information and Event Management. [online] Available at: https://gartner.com/doc/reprints?id=1-4LC8PAW&ct=171130&st=sb [Accessed 16 Dec. 2017].

Elastic.co, (2017a). Elasticsearch Reference 6.0. [online] Available at: https://www.elastic.co/guide/en/elasticsearch/reference/6.0/index.html [Accessed 6 Dec. 2017]

Tyler Lutz, (2017). OSSEC & ELK Stack Integration [online] Available at: https://practicalassurance.com/blog/ossec-elk-stack-integration/ [Accessed 2 Feb. 2018]

Elastic.co, (2017b). Logstash Reference 6.0. [online] Available at: https://www.elastic.co/guide/en/logstash/6.0/index.html [Accessed 6 Dec. 2017]

Elastic.co, (2017c). Kibana User Guide 6.1. [online] Available at: https://www.elastic.co/guide/en/kibana/6.1/index.html [Accessed 17 Dec. 2017]

Elastic.co, (2017d). Beats Platform Reference. [online] Available at: https://www.elastic.co/guide/en/beats/libbeat/6.1/index.html [Accessed 17 Dec. 2017]

Elastic.co, (2017e). X-Pack Subscriptions. [online] Available at: https://www.elastic.co/subscriptions [Accessed 24 Dec. 2017]

Floragunn.com, (2017). Search Guard. [online] Available at: https://floragunn.com/#searchguard_overview [Accessed 24 Dec. 2017]

PCI Security Standards Council, LLC (2016). Requirements and Security Assessment Procedures 3.2. [online] Available at: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2.pdf [Accessed 16 Feb. 2018]

Gormley C. & Tong Z. (2015). Elasticsearch: The Definitive Guide. [online] Available at: https://www.elastic.co/guide/en/elasticsearch/guide/2.x/index.html [Accessed 16 Dec. 2017]

Lonvick, C. (2001). The BSD Syslog Protocol (RFC 3164). [online] Available at: https://datatracker.ietf.org/doc/rfc3164/?include_text=1 [Accessed 25 Dec. 2017]

Reynolds, R. (2016). Package management on Windows with Chocolatey & Puppet. [online] Available at: https://puppet.com/blog/package-management-windows-chocolatey-puppet [Accessed 25 Dec. 2017]

Atanasov, M. (2017). Killing your Logstash performance with Grok. [online] Available at: https://medium.com/@momchil.dev/killing-your-logstash-performance-with-grok-f5f23ae47956 [Accessed 28 Dec. 2017]

Rao, S. (2015). Add support for clustering Logstash instances #2632. [online] Available at: https://github.com/elastic/logstash/issues/2632 [Accessed 7 Jan. 2018]

Carey, J. (2017). TLS for the Elastic Stack: Elasticsearch, Kibana, Beats and Logstash. [online] Available at: https://www.elastic.co/blog/tls-elastic-stack-elasticsearch-kibana-logstash-filebeat [Accessed 5 Feb. 2018]

Elastic.co (2017f). Subscriptions. [online] Available at: https://www.elastic.co/subscriptions [Accessed 5 Feb. 2018]

M-Files Oy. (2017). The M-Files COM/.NET API. [online] Available at: http://developer.m-files.com/APIs/COM-API/ [Accessed 14 Jan. 2018]

Mozilla. (2018). XPath. [online] Available at: https://developer.mozilla.org/en-US/docs/Web/XPath [Accessed 14 Jan. 2018]

Ondas, R. (2017). Force_array is not applied correctly. #46. [online]. Available at: https://github.com/logstash-plugins/logstash-filter-xml/issues/46 [Accessed 14 Jan. 2018]

Source code (C#) of the M-Files log extractor prototype

```
using System;
using System.Collections;
using System.Xml;
using MFaaP.MFilesAPI;
using MFilesAPI;

namespace MFilesLogExtractor
{
    class Program
    {

        // Hashtable for M-Files vaults
        static Hashtable vaultProperties = new Hashtable
            {
                { "{6F3BC1C0-F6E6-439B-8973-38738C884E7E}", 1148 }, // Sample
                { "{52EDDFD6-CCF2-44BA-8B5D-65370D074F45}", 1149 } // My Vault
            };

        static void Main(string[] args)
        {

            // Extract event logs from all vaults
            foreach (DictionaryEntry vaultProperty in vaultProperties)
            {
                // Establish vault connectivity
                (Vault vault, MFilesServerApplication application) =
ConnectVault(vaultProperty.Key.ToString());

                // Extract event logs
                String rawEventLog = ExtractEventLog(vault);

                // Enrich event log
                XmlDocument xmlEventLog = EnrichXml(rawEventLog, vault,
int.Parse(vaultProperty.Value.ToString()));

                // Save event log
                SaveEventLog(xmlEventLog, vaultProperty.Key.ToString());

                // Disconnect from M-Files
                application.Disconnect();
            }
        }

        static (Vault, MFilesServerApplication) ConnectVault(String GUID)
        {
            // Assign variables
            Vault vault = null;
            MFilesServerApplication application = null;

            // Establish connection
            var connectionDetails = new ConnectionDetails();

            // Connect to vault
            try
            {
                connectionDetails.ConnectToVaultAdministra-
tive(Guid.Parse(GUID), out vault, out application);
            }
            catch (Exception err)
            {
                Console.Write(err);
            }
```

```
            // Return vault and application objects
            return (vault, application);
        }

        static String ExtractEventLog(Vault vault)
        {
            // Retrieve event log server
            String eventLog = vault.EventLogOperations.ExportAll();

            // Clear event log from server
            // vault.EventLogOperations.Clear();

            return eventLog;
        }

         static void SaveEventLog(XmlDocument eventLog, String GUID)
        {
            // Create Directory
            String directoryPath = Environment.GetFolderPath(Environment.Spe-
cialFolder.CommonApplicationData) + "\\MFilesEvents";
            System.IO.Directory.CreateDirectory(directoryPath);

            // Save the events
            eventLog.Save(@directoryPath + "\\eventLog_" + GUID + ".xml");
        }

        static XmlDocument EnrichXml(String eventLog, Vault vault, int gdpr-
property)
        {
            // Load XML
            XmlDocument xmlDoc = new XmlDocument();
            xmlDoc.LoadXml(eventLog);

            // Choose only nodes which has the objver child
            XmlElement xmlRoot = xmlDoc.DocumentElement;
            XmlNodeList xmlNodes = xmlRoot.SelectNodes("event/data/objectver-
sion/objver");

            // If there is no objver child, skip iteration
            if (xmlNodes != null)
            {
                // Iterate
                foreach (XmlNode xmlNode in xmlNodes)
                {

                    // Get object version
                    int objVer;
                    try
                    {
                        objVer = int.Parse(xmlNode["version"].InnerText);
                    } catch
                    {
                        objVer = 0; // Set to zero if version does not exist
in xml file
                    }

                    // Get Object ID and type
                    int objId = int.Parse(xmlNode["objid"].InnerText);
                    int objType = int.Parse(xmlNode["objtype"].Attrib-
utes["id"].Value);

                    // Check the document for GDPR relation
                    String relation = CheckGdprRelation(objType, objVer, ob-
jId, vault, gdprproperty);

                    // Create new xml element and append existing doc
```

```
                    XmlElement xmlElement = xmlDoc.CreateElement("gdprrela-
tion");
                    xmlElement.InnerText = relation;
                    xmlNode.AppendChild(xmlElement);

                }
            }

            return xmlDoc;
        }

        static String CheckGdprRelation(int objTypeId, int version, int ob-
jectID, Vault vault, int gdprProperty)
        {

            // Convert variables to m-files objects
            ObjVer objVer = new ObjVer();
            objVer.SetIDs(objTypeId, objectID, version);

            // Load object properties and check for GDPR relation
            try
            {
                ObjectVersionAndProperties objectProperties = vault.ObjectOp-
erations.GetObjectVersionAndProperties(objVer, true);
                String value = objectProperties.Properties.SearchForProp-
erty(gdprProperty).GetValueAsText(true, true, true, true, true, true);

                // Check if the value is empty
                if (value != null)
                {
                    return value;
                }

            } catch (Exception err)
            {
                Console.Write(err);
            }

            return "Not defined";
        }

    }
}
```

Logstash pipeline configuration file for M-Files event logs

```
input {
    beats {
        port => 5044
    }
}

filter {
    # Perform filtering only for m-files beats
    if ([fields][beat_source] == "mfiles") {

        # Parse XML for document fields
        xml {
            source => "message"
            store_xml => "false"
            xpath => [
                # Extract relevant events to separate fields
                "/event/id/text()", "mf.eventid",
                "/event/type/text()", "mf.eventtype",
                "/event/timestamp/text()", "mf.timestamp",
                "/event/causedbyuser/@loginaccount", "mf.loginaccount",
                "/event/data/objectversion/objver/objid/text()", "mf.ob-
jectid",
                "/event/data/objectversion/objver/version/text()", "mf.ob-
jectversion",
                "/event/data/objectversion/objver/gdprrela-
tion/text()", "mf.gdpr",
                "/event/data/objectversion/origina-
lobjid/vault/text()", "mf.vaultguid",
                "/event/data/objectversion/title/text()", "mf.title"
            ]
        }

        # Workaround for Logstash issue #46 (Force_array is not applied cor-
rectly)
        mutate {
            # Convert results from array to string
            replace => { "mf.eventid" => "%{mf.eventid[0]}" }
            replace => { "mf.eventtype" => "%{mf.eventtype[0]}" }
            replace => { "mf.timestamp" => "%{mf.timestamp[0]}" }
            replace => { "mf.loginaccount" => "%{mf.loginaccount[0]}" }
        }

        # Check if object specific fields exists and convert if they do
        if [mf.objectid] {
            mutate {
                replace => { "mf.objectid" => "%{mf.objectid[0]}" }
            }
        }
        if [mf.objectversion] {
            mutate {
            replace => { "mf.objectversion" => "%{mf.objectversion[0]}" }
            }
        }
        if [mf.gdpr] {
            mutate {
            replace => { "mf.gdpr" => "%{mf.gdpr[0]}" }
            }
        }
        if [mf.vaultguid] {
            mutate {
            replace => { "mf.vaultguid" => "%{mf.vaultguid[0]}" }
            }
```

```
        }
        if [mf.title] {
            mutate {
            replace => { "mf.title" => "%{mf.title[0]}" }
            }
        }

        # Match for date
        date {
            match => [
                "mf.timestamp", "yyyy-MM-dd HH:mm:ss.SSS'000000'"
            ]
            timezone => "Europe/Helsinki"
            target => "@timestamp"
        }

        # Remove unnecessary fields
        mutate {
            remove_field => [
                "mf.timestamp",
                "source",
            ]

            remove_tag => [
                "beats_input_codec_plain_applied"
            ]
        }
    }
}

# Output to Elasticsearch
output {
    elasticsearch {
        hosts => "localhost:9200"
        manage_template => false
        index => "siem"
        document_type => "test"
    }

    stdout {
        codec => rubydebug
    }

}
```